基于加权的无人机集群组网分簇算法 *

王沁飞，南建国，黄金科，王　彪，贾镇泽

(空军工程大学 航空工程学院, 西安 710038)

摘　要： 在无人机集群组网中，节点的高速移动会造成网络拓扑结构更新频繁，使网络管理变得更加复杂。分簇能够增大网络容量，实现空间资源的复用，是优化网络管理的有效手段之一。针对大规模、高速移动的环境进行了研究，提出了一种多参数加权分簇算法。该算法将最大速度相似度分簇算法中的分簇指标引入到加权分簇算法中，并且对链路保持率、节点度差、节点剩余能量进行改进，综合考虑这四种参数，通过加权组合的方式选举具有最大权重的网络节点作为簇头。仿真结果表明，该分簇算法不仅能够减少簇的数量和簇间切换率，提高分簇的稳定性，而且能够延长最小节点生存时间，改善网络的整体续航能力。

关键词： 无人机集群组网；分簇算法；最大速度相似度；链路保持率；节点度；剩余能量

中图分类号： TP393.2　　doi: 10.3969/j.issn.1001-3695.2017.12.0764

Weighting based clustering algorithm for FANET

Wang Qinfei, Nan Jianguo, Huang Jinke, Wang Biao, Jia Zhenze

(*Aeronautics Engineering College*, *Air Force Engineering University*, *Xi'an* 710038, *China*)

Abstract: In FANET, the high-speed movement of nodes will result in frequent updates of the network topology, which makes network management more complicated. Clustering can increase network capacity and reuse spatial resources, which is an effective way to optimize network management. In the large-scale and high-speed mobile environment, this paper proposes a multi-parameter weighted clustering algorithm. This algorithm introduces the clustering index in the maximum speed similarity algorithm into the weighted clustering algorithm, and improves the link retention, node degree difference and node residual energy to select the network node with largest weight as the cluster head by considering these four parameters synthetically. The simulation results show that the clustering algorithm can reduce the number of clusters and the switching rate between clusters, which improves the stability of clustering, and can extend the minimum survival time of nodes, which improves the overall endurance of the network.

Key words: FANETs; clustering algorithm; maximum speed similarity; link retention; node degree; residual energy

0　引言

随着人工智能、传感器以及通信技术的快速发展，人类已经制造出能够自主飞行或远程操控的无人飞行器[1]（unmanned aerial vehicle，UAV）。由于无人机具有全寿命周期费用低、结构简单、隐蔽性好、起降灵活等优点，人们将大量的无人机按照集群化的方式应用到军事、民用领域，并获得了良好的效果。无人机集群自组织网络（flying Ad-hoc networks，FANET）是移动自组织网络（mobile Ad-hoc networks，MANET）在无人机领域的典型应用，它不仅具备移动自组织网络无中心、自组织、动态拓扑的特点，同时也面临着新的挑战，比如高动态下拓扑结构更新频繁、节点能量有限、脆弱的网络安全等[2-4]。在大规模以及高速移动的网络情况下，网络管理变得尤为困难，而分簇是优化网络管理的有效手段之一。

目前在无线自组网领域，许多学者提出了不同的分簇算法，包括最高节点度分簇算法、最小 ID 分簇算法、最低移动性分簇算法、加权分簇算法等。前四种分簇算法只考虑单方面因素，无法适应更多样化的无线自组网的应用场景。最高节点度分簇算法[5]是基于连接度进行分簇的，通过计算节点周围的邻居节点数目，选取具有合适邻居节点数目的节点做为簇头。最小 ID 分簇算法[6]是 Lin 和 Gerla 在 1995 年最早提出的，在网络中每个节点都对应唯一的 ID 号，通过选择相邻节点中 ID 号最小的节点作为簇头。最低移动性分簇算法[7]是选择相邻节点中移动性最小的节点做为簇头。加权分簇算法[8]（weighted clustering

基金项目： 国家自然科学基金资助项目

作者简介： 王沁飞（1993-），男，河南沁阳人，硕士研究生，主要研究方向为通信与网络技术（641808277@qq.com）；南建国（1969-），男，教授，硕士，主要研究方向为机载计算机技术；黄金科（1988-），男，博士研究生，主要研究方向为无线自组网技术；王彪（1992-），男，博士研究生，主要研究方向为目标探测与侦察监视；贾镇泽（1989-），男，博士研究生，主要研究方向为目标识别技术.

algorithm，WCA）与上述几种算法不同，它不是考虑单一方面因素，而是综合多方面因素，比如节点度、移动性、剩余能量、节点的邻居节点位置等，相较于考虑单方面因素的分簇算法来说，能更好地适应不同的应用场景。

目前，学者对于 WCA 算法的改进算法有很多，比如兰州交通大学的丁青提出的基于权值的自适应分簇算法 NWBACA[9]，综合考虑节点的剩余能量、节点度、节点与邻居节点的平均距离、节点的移动速度等四个方面因素进行簇头选举，增强了网络的稳定性和整网的生存时间，但是在考虑能量时仅仅考虑节点作为簇头时，消耗的能量与节点度的关系，没有考虑节点作为普通节点时能量的消耗也与节点度有关。在计算移动性时，考虑的是节点的移动速度，不足以说明节点与其邻居节点的运动相似性。赛塔特科技大学 LAVETE 实验室的 Dyabi 等人提出的基于节点性能的加权分簇算法[10]中，虽然运用了加权分簇的思想，但是侧重于节点的运存、CPU、硬盘等性能，对节点的能量、移动性等方面考虑的不多。重庆大学的唐丰等人提出的多参数加权分簇算法[11]，综合考虑节点剩余能量、邻居节点参数通过计算节点的移动性参数和综合稳定参数进行簇头的选择，与仅考虑单一因素的 HD 分簇算法和 MOBIC 分簇算法进行对比，网络分簇简洁、稳定，但是该算法在考虑节点剩余能量时，没有给出具体的能量消耗的计算方式，在计算节点移动性参数时，只考虑节点的运动速度和方向，没有考虑节点与邻居节点的运动相似性。郑州大学的马豫青等人提出的自适应安全加权分簇算法 SWCA[12]，综合考虑节点与其邻居节点的连通性、能量、移动性、节点安全性，侧重于网络的安全性方面，而对于节点的能量、移动性等方面没有做进一步的改进。本文通过对相关文献的研究，提出了一种改进的多参数组合加权分簇算法（improved-weighted clustering algorithm，IWCA），综合考虑节点的最大速度相似度、平均链路保持率、自适应节点度差、节点剩余能量四个参数，通过四个参数加权组合，选举具有最大权重的网络节点作为簇头。在参数的选择上，最大速度相似度能够更加直观的显示出该节点与其邻居节点的运动相似性，所以本文将最大速度相似度分簇算法[13]引入到组合加权分簇算法中，但是在计算相似度时不是通过速度和方向的方式进行计算的，而是将速度进行分解，通过计算不同方向的平均速度差和速度差方差，来体现节点与邻居节点的速度相似度。自适应节点度差较之前的节点度差[14]来说，主要区别在于理想节点度的计算方法上，本文采用平均节点度来代替理想节点度，使网络能够根据不同的拓扑结构自适应调整分簇的标准。节点的平均距离没有考虑节点之后的运动，虽然最大速度相似度有考虑速度的影响，但是对于无人机集群组网来说，高移动性导致的拓扑结构更新频繁是影响网络性能最主要的因素，因此，在原有链路保持时间[15]的基础上增加求平均值过程，来替换原有的平均节点距离，不仅能够体现出距离，还能预测节点与节点之间的链路保持时间，更适合无人机集群组网。无人机集群飞行时，如果一个节点长时间担任簇头，能量消耗速度加快，这将会降低整网生存时间，使该节点提前退出战斗，这对于作战或者其他应用场景来说是不愿意看到的，因此，在选举簇头时，必须考虑节点的剩余能量，但在计算节点剩余能量时，不仅需要考虑节点担任簇头时簇内成员的数量对于能量消耗的影响，如文献[9]，还需要考虑节点担任簇内普通节点时，邻居节点数量对能量消耗的影响。为了解决不同参数单位不同的问题，统一对四个参数进行归一化处理，避免了某一参数起主导作用所引起的分簇不公平现象。

1 分簇联合度量指标

1.1 改进的最大速度相似度

在传统的组合加权分簇算法中，在计算节点移动性时主要考虑节点速度差、平均移动速度等，这些参数无法体现节点与其邻居节点的运动相似性。在节点的移动性与其邻居节点的移动性相差很大时，很容易导致网络拓扑结构改变，导致二次分簇。重新分簇会导致较大的网络资源浪费，增加路由开销，影响网络性能。所以，将最大速度相似度分簇算法中的最大速度相似度参数引入到加权分簇算法中来表示节点的移动性，可以很好的避免这个问题。在传统的最大速度相似度分簇算法中，利用速度和方向两个参数进行计算，速度的计算相对简单，但方向角度的计算存在一定的困难。本文提出一种利用节点的速度信息和位置信息在二维坐标下计算节点最大速度相似度的方法。

如图 1 所示，节点 O 与节点 A 在 x、y 轴方向上的速度差分别为：

$$\begin{aligned} v_{Ax} &= v_O \cos\alpha - v_A \cos\beta \\ v_{Ay} &= v_O \sin\alpha - v_A \sin\beta \end{aligned} \tag{1}$$

假设节点 O 有 n 个邻居节点，则节点 O 与周围邻居节点的 x、y 轴方向的平均速度差为：

$$\begin{aligned} \bar{v}_{Ox} &= \frac{\sum_{i=1}^{n}(v_O\cos\alpha - v_i\cos\theta_i)}{n} \\ \bar{v}_{Oy} &= \frac{\sum_{i=1}^{n}(v_O\sin\alpha - v_i\sin\theta_i)}{n} \end{aligned} \tag{2}$$

根据勾股定理和式（2），可以计算出节点 O 与 n 个邻居节点的平均速度差：

$$\bar{v}_O = \sqrt{\bar{v}_{Ox}^{\,2} + \bar{v}_{Oy}^{\,2}} \tag{3}$$

要想体现出速度差相似度，必须求得节点 O 与其邻居节点的 x、y 轴速度差的方差：

$$\begin{aligned} \sigma_{Ox}^2 &= \frac{1}{n}\sum_{i=1}^{n}(\bar{v}_{Ox} - v_{Oix})^2 \\ \sigma_{Oy}^2 &= \frac{1}{n}\sum_{i=1}^{n}(\bar{v}_{Oy} - v_{Oiy})^2 \end{aligned} \tag{4}$$

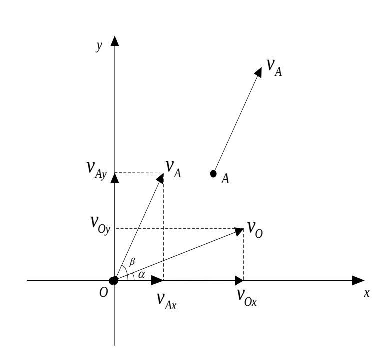


图 1　某一时刻节点 O 与节点 A 的运动状态

在式（4）中，v_{Oix} 表示节点 O 与节点 i 在 x 轴方向上的速度差，v_{Oiy} 表示节点 O 与节点 i 在 y 轴方向上的速度差。将 x、y 轴方向上的速度差方差进行组合加权，节点 O 与邻居节点的最大速度相似度用 σ_O^2 表示为

$$\sigma_O^2 = \frac{\sigma_{Ox}^2 + \sigma_{Oy}^2}{2} \quad 5）$$

速度差体现了节点 O 与其邻居节点速度的相似度；方差是为了剔除坏值，防止出现极端。将平均速度差和速度差方差两个参数整合在一起，作为衡量移动性的指标。随着节点之间速度差与方差的增大，节点的最大速度相似度越小。最后在归一化处理之后，改进的最大速度相似度用公式表示为

$$M_O = e^{-(\sigma_O^2 + \bar{v}_O^2)} \quad（6）$$

1.2　平均链路保持率

平均链路保持时间不仅能够体现出节点与邻居节点的距离，而且可以对邻居节点速度大小和运动方向进行预测，从而预测最大链路保持时间。为简化计算，这里假设节点在一段时间内运动方向和速度大小保持不变。

如图 2 所示，r 为节点 A 的最大通信半径，A、B 的位置信息可以由 GPS 或其他定位系统获得，节点 A 在 Δt 时间内连续两次获得节点 B 的位置，在假设节点 B 在一段时间里的运动速度和方向保持不变的前提下，可以估算出节点 B 的运动速度与运动方向，速度如式（7）所示。

$$v_B = \frac{\sqrt{\left(x_B' - x_B\right)^2 + \left(y_B' - y_B\right)^2}}{\Delta t} \quad（7）$$

其中：节点 B 在 t 时刻的坐标为 (x_B, y_B)，在 $t+\Delta t$ 时刻的坐标为 (x_B', y_B')，节点 B 的运动方向沿着 (x_B, y_B) 到 (x_B', y_B') 的方向，如图 2 所示。

在此基础上，假定节点 A 静止，则节点 B 相对于节点 A 的速度大小和方向如图 2 所示。由 $\vec{v}_B^* = \vec{v}_A + \vec{v}_B$ 可以求得直线 CD 的斜率，假设为 k，已知 B 点坐标，可以求得直线 CD 在直角坐标系中的方程如下：

$$y_{CD} = k*(x - x_B) + y_B \quad（8）$$

根据点到直线的距离公式，代入式（8）中，可得节点 A 与直线 CD 的距离 d_{AE}：

$$d_{AE} = \frac{\left|k\left(x_A - x_B\right) + y_B - y_A\right|}{\sqrt{k^2+1}} \quad（9）$$

根据几何关系，可以求得节点 A 与节点 B 之间的距离 d_{AB}、点 C 与点 E 之间的距离 d_{CE}、节点 B 与点 E 之间的距离 d_{BE}。由距离与速度的关系，可以推测节点 B 与节点 A 的链路保持时间 T_{AB} 为

$$T_{AB} = \frac{d_{CE} \pm d_{BE}}{\left|\vec{v}_B^*\right|} \quad（10）$$

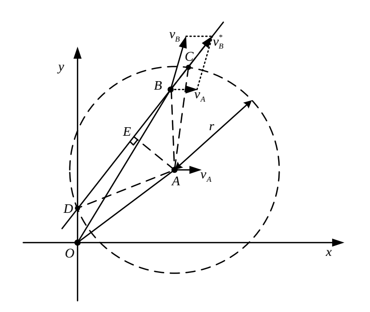


图 2　节点 A 与节点 B 在某一时刻的运动状态

由于 $0 \le d_{BE} \pm d_{CE} \le 2r$，所以节点 B 与节点 A 的能够取到的最大链路保持时间为

$$T_{\max}^B = \frac{2r}{\left|\vec{v}_B^*\right|} \quad（11）$$

知道直线 CD 的斜率，可以求得直线 AE 的斜率，已知 A 点的坐标，不难求出 E 点的坐标，设为 (x_E, y_E)。在 $k>0$ 的情况下，$x_B > x_E, \cos\alpha > 0$ 或 $x_B < x_E, \cos\alpha < 0$ 时公式中符号取减号，否则取加号；在 $k<0$ 的情况下，$x_B < x_E, \cos\alpha < 0$ 或 $x_B > x_E, \cos\alpha > 0$ 时公式中符号取减号，否则取加号；其中 α 为 $\vec{v}_B^*$ 与 x 轴正向的夹角。

通过上述方法，可以求得平均链路保持时间和平均最大链路保持时间：

$$T_A = \frac{\sum_{i=1}^{n} T_{Ai}}{n} \quad（12）$$

$$T_{\max} = \frac{\sum_{i=1}^{n} T_{\max}^i}{n} \quad（13）$$

链路保持时间越大，说明网络的拓扑结构越稳定，簇的更新频率就会降低，所以进行归一化处理后，平均链路保持率为

$$P_A = \frac{T_A}{T_{\max}} \quad（14）$$

1.3　自适应节点度差

由于网络带宽资源的有限，节点周围邻居节点的数量应该控制在一定的范围内。节点度大，会减少簇的数量，但可能会造成网络拥堵，影响服务质量；节点度小，会对宝贵的网络带宽资源造成严重的浪费。所以，本文在节点度差的基础上通过改进比较对象，使用平均节点度来代替网络理想节点度[14]，这样能够使网络节点分布较均匀，既可以对带宽资源的充分利用，

又不至于造成网络拥塞。另外，用平均节点度来代替理想节点度，能够适应多种场合。

在网络中，每个节点通过交互 Hello 消息来获取本节点的节点度，节点 i 的节点度的定义为

$$d_i = \sum_{j=0}^{n} d_j^i \tag{15}$$

其中：

$$d_j^i = \begin{cases} 1 & 0 < d_{ij} < r \\ 0 & d_{ij} \geq r \end{cases} \tag{16}$$

式（16）中，d_{ij} 为节点 i 与节点 j 之间的距离。

平均节点度是所有节点的节点度的总和除以节点数量，假设有 n 个网络节点，则平均节点度的计算公式为

$$\bar{d} = \frac{\sum_{i=0}^{n} d_i}{n} \tag{17}$$

用每个节点的节点度减去平均节点度，得到自适应节点度差为

$$\varepsilon_i = \left| d_i - \bar{d} \right| \tag{18}$$

在网络中，节点度差越小，说明网络节点的分布越平均。经过归一化处理后，节点 i 的自适应节点度差为

$$D_i = e^{-\varepsilon_i} \tag{19}$$

1.4 节点剩余能量

在分簇算法中，大多数加权分簇算法在计算节点剩余能量时只考虑节点充当簇头的时间，计算时忽略节点充当簇头时网络的负载量，更很少有人考虑在节点为簇内成员时，网络负载量对节点能量消耗的影响。本文对节点剩余能量计算方法进行改进，考虑节点充当簇头和普通节点时，节点度对能量消耗的影响，使计算方法更加合理。由于网络拓扑结构不断变化，所以节点的邻居节点数也是不断变化的，不能简单将节点担任簇头的时间相加。

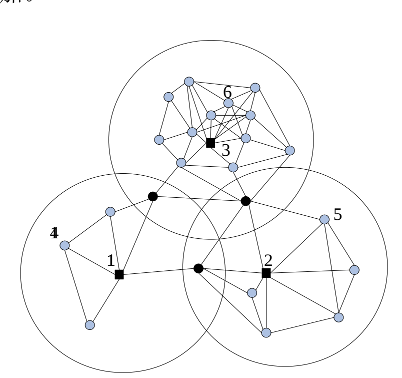


图 3　某一时刻网络部分簇结构示意图

如图 3 所示，在节点 1、2、3 担任簇头期间，由于其邻居节点数量不等，所以消耗能量也是不相同的，节点 4、5、6 为普通节点时，由于其邻居节点数量不等，所以能耗也不能认为是一样的。假设节点初始能量为 E，剩余能量为 E_l，节点在担任普通节点时，单位时间内单位节点度消耗的能量为 e_1，在担任簇头时，单位时间内单位节点度消耗的能量为 e_2，则节点在一段工作时间后的剩余能量为

$$E_l = E - \sum_{1}^{i} e_1 d_{ni} t_i - \sum_{1}^{j} e_2 d_{nj}^{*} t_j \tag{20}$$

其中：i 为节点担任普通节点的次数，d_{ni} 为节点第 i 次担任普通节点的节点度，t_i 为节点第 i 次担任普通节点的时间；j 为节点担任簇头的次数，d_{nj}^{*} 为节点第 j 次担任簇头的节点度，t_j 为节点第 j 次担任簇头的时间。

节点剩余能量越高，越适合被选为簇头，所以，将剩余能量除以总能量，用百分比进行归一化处理。衡量节点剩余能量的参数为 η_i，用公式表示如下：

$$\eta = \frac{E_l}{E} \tag{21}$$

1.5 联合度量指标

由 1.1~1.4 节可以得到节点 i 的最大速度相似度参数 M_i、平均链路保持率参数 P_i、自适应节点度差参数 D_i、节点剩余能量参数 η_i，综合这四个参数，节点 i 选举簇头的权重参数为

$$W_i = w_1 M_i + w_2 P_i + w_3 D_i + w_4 \eta_i \tag{22}$$

其中：w_1、w_2、w_3、w_4 分别为四种参数的权值，且 $w_1 + w_2 + w_3 + w_4 = 1$，具体取值，可以根据实际的应用场景进行设置。

2 分簇算法实现

2.1 簇初始化

本文提出的 IWCA 分簇算法，综合考虑节点的最大速度相似度、平均链路保持率、自适应节点度差和剩余能量四个参数，通过加权组合选举具有最大权重的节点作为簇头。具体实现步骤如下：

a)网络创建初期，所有网络节点都处于初始状态。节点通过周期性广播 Hello 消息，来获得邻居节点的状态信息。节点 i 的所有一跳邻居节点（位于节点 i 的通信半径之内的节点均称为一跳邻居节点）的数目为节点度 d_i。

b)通过收集邻居节点的节点度，根据式（17）计算平均节点度 $\bar{d}$，通过式（18）(19）计算该节点的自适应节点度差参数 D_i。

c)节点通过接收邻居节点的 Hello 消息，获得邻居节点的位置信息，通过连续两次获得的位置信息，估计邻居节点的运动速度。根据式（1）(6）计算节点的最大速度相似度参数 M_i。

d)根据节点与邻居节点的相对位置以及运动方向，求得运动轨迹的斜率。在 $k>0$ 的情况下，$x_B > x_E, \cos\alpha > 0$ 或 $x_B < x_E, \cos\alpha < 0$ 时式（10）符号取减号，否则取加号；在 $k<0$ 的情况下，$x_B < x_E, \cos\alpha < 0$ 或 $x_B > x_E, \cos\alpha > 0$ 时式（10）符号取减号，否则取加号；由式（7）(14）求得节点 i 的平均链路保持率 P_i。

e)根据式（20），估计节点 i 的剩余能量，求得节点剩余能量参数 η_i。

f)用之前设置好的权重值乘以计算出来的参数值，求和得

到节点 i 自身的权重因子 W_i。通过与邻居节点的权重因子进行比较，选择具有最大权重因子的节点作为簇头。

g)重复步骤 a)~f)，直到所有节点的状态确定完毕。

2.2 簇维护机制

在网络进行分簇时，剧烈的节点移动会使节点经常加入或者离开簇，更有可能导致簇头的更新和网络重新配置结构，这会造成较大的路由开销，增大计算负载。所以必须有合理的簇维护机制来尽量减少分簇开销，维持簇的稳定。

簇维护机制包括以下四种规则：

a）删除簇内节点。当一个成员节点收不到带有簇头 ID 信息的 Hello 消息或者簇头节点收不到该节点周期性广播的 Hello 消息时，表明该节点已经不在簇头的最大通信范围内，应该将该节点信息从簇内删除。

b）增加新节点。新节点包括网络中最新加入的节点和被其他分簇删除的节点。当一个新节点想加入一个分簇时，通过“握手”行为确定关系，并重新计算自身权重进行簇头的竞选。

c）更换簇头。当簇头收不到带有簇内成员 ID 信息的 Hello 消息或者簇头因为其他原因退出网络时，视为放弃簇头的“领导地位”。此时，簇内成员恢复到初始状态，按照分簇的七个步骤重新进行簇头的选举。

d）簇的合并。当两个簇的簇头成为了一跳邻居节点时，面临簇的合并问题。比较两个簇头与簇内成员的最大速度相似度，如果两者相近，说明两个簇会在各自的范围内停留很久，此时比较两个簇的权重，选举权重大的为新的簇头，之后该簇头周期性广播簇头信息，确定自己的“领导地位”。如果两个簇头与簇内成员的最大速度相似度相差很大，说明两个簇只是“擦肩而过”，若合并为一个簇，会造成较大的网络开销。

3 仿真与性能分析

本文利用 MATLAB 仿真软件对 IWCA 进行仿真分析。各节点在 1000*1000 单位内按照高斯-马尔可夫移动模型自由移动，节点的最大通信距离为 200 个单位长度。节点初始能量为 1，作为簇头时，单位时间内单位邻居节点之间通信能量消耗为 0.002，作为普通节点时，单位时间内单位邻居节点之间通信能量消耗为 0.001。仿真时间设置为 30 min。

考虑到该分簇算法是应用在无人机集群组网中，移动性和剩余能量的影响比较大，所以参数重要性排序为 $M_i > D_i > \eta_i > P_i$，根据质心权重[16~18]的权值计算方法，计算的权重结果为 0.5225、0.27、0.145、0.0625。

由于分簇算法中对于四个参数的改进，体现在移动性、节点度、剩余能量、平均链路保持时间上，所以选用簇的数量、簇间切换次数、网络最小节点生存时间三个指标来衡量分簇算法优劣。并与现有的 MOBIC、AOWLR、NWBACA、WCA 分簇算法进行对比分析。其中网络最小节点生存时间是指单次仿真所有节点中生存时间最短的节点的生存时间。

图 4 反映了节点数量对平均簇头数的影响。随着网络中节点数量的增大，IWCA、MOBIC、AOWLR、NWBACA、WCA 算法的平均簇头数都在不断增大，但 IWCA 对理想节点度的改进，既不会使簇内成员相对较少，造成带宽资源的浪费，也不会使簇内成员相对较多，造成网络的拥堵，平均簇头数均低于其他四种分簇算法。

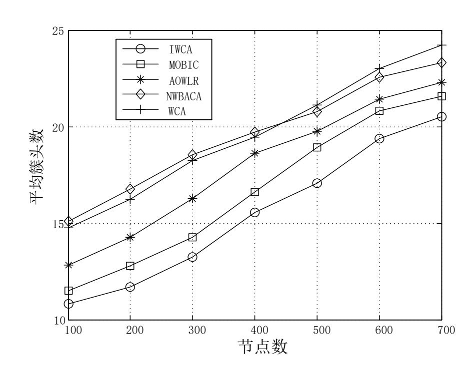


图 4　节点数量对簇头数的影响

定义单位时间内簇间切换次数除以网络节点的总数为簇间切换率。仿真设置网络节点数为 200 个。

图 5 反映的是节点移动速度对分簇结构稳定性的影响。随着网络中节点移动速度的不断增大，簇间切换率是呈上升趋势的即簇间切换次数是呈上升趋势的。这是因为网络节点的高移动性，会导致网络拓扑结构的更新速度加快和通信链路的断开与连接，从而使簇的更新更加频繁。由于该算法综合考虑节点的最大速度相似度和平均链路保持时间，所以 IWCA 算法的簇间切换率比其他四种分簇算法低，具有更好的稳定性。

图 6 反映的是节点数对网络最小节点生存时间的影响。如图 6 所示，在同一节点数目下，IWCA 算法的节点最小生存时间比其他四种分簇算法的大。节点最小生存时间比较大相当于提高了整网生存时间的“最短板”，这是因为 IWCA 分簇算法中考虑到节点的剩余能量，而且权重设置的比较大，因为不仅考虑到节点作为簇头时消耗的能量与簇内成员数目的关系，而且考虑到节点作为普通节点时消耗的能量与其邻居节点数目的关系，所以，当节点的剩余能量较小时，其权重参数计算结果会比较小，很难竞争成为簇头。通过这种方式对该节点进行保护，减少该节点的能量消耗，这样可以增大网络整体的续航能力，增强网络的鲁棒性。

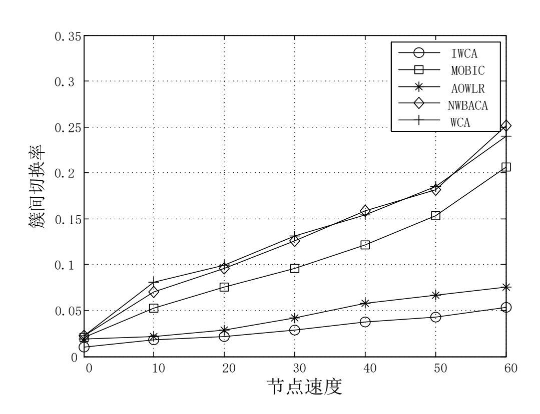


图 5　节点移动速度对分簇结构稳定性影响

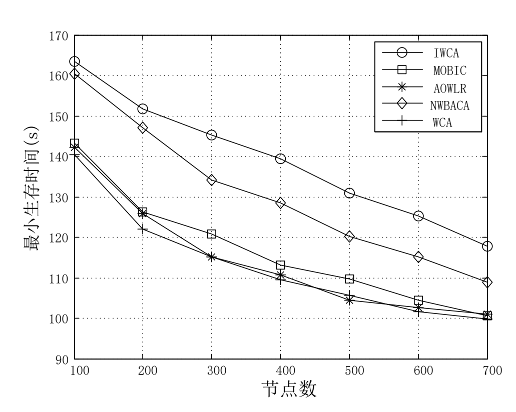


图 6　节点数对网络节点生存时间的影响

（节点最大移动速度为 20m/s）

4　结束语

无人机集群组网是移动自组网在无人机领域的一种典型应用。无人机集群组网不仅具有移动自组网的特点，还要面临更加严峻的节点集群化、节点高速移动的问题。本文以 WCA 算法为基础，在体现移动性方面引入最大速度相似度分簇算法，将距离改进为平均链路保持时间，对节点度差和节点剩余能量分别进行改进，使计算结果更加科学，提高了分簇的稳定性，增大了整网的续航能力。

参考文献：

[1] Bekmezci I, Sahingoz O K, Temel S. Flying Ad-hoc networks (FANETs): a survey [J]. Ad Hoc Networks, 2013, 11 (3): 1254-1270.

[2] Zafar W, Khan B M. Flying Ad-hoc networks: technological and social implications [J]. IEEE Technology & Society Magazine, 2016, 35 (2): 67-74.

[3] Yu J Y, Chong P H J. A survey of clustering schemes for mobile Ad-hoc networks [J]. IEEE Communications Surveys & Tutorials, 2005, 7 (1): 32-48.

[4] Sreevatsan A P, Thomas D. An optimal weighted cluster based routing protocol for MANET [C]// Proc of International Conference on Data Mining & Advanced Computing. New York: IEEE Press, 2016: 310-316.

[5] Tselikis C, Mitropoulos S, Douligeris C, et al. Empirical study of clustering algorithms for wireless Ad hoc networks [C]// Proc of International Conference on Systems Signals and Image Processing. New York: IEEE Press, 2009: 1-6.

[6] Gerla M, Tsai T C. Multicluster, mobile, multimedia radio network [J]. Wireless Network, 1995, 1 (3): 255-265.

[7] Ni M, Zhong Z, Zhao D. MPBC: a mobility prediction-based clustering scheme for Ad hoc networks [J]. IEEE Trans on Vehicular Techbolog, 2011, 60 (9): 4549-4559.

[8] Chatterjee M, Das S K, Turgu D t. WCA: a weighted clustering algorithm for mobile Ad hoc networks [J]. Clustering Computing, 2002, 5 (2): 193-204.

[9] 丁青. 一种新的 Ad hoc 网络自适应分簇算法 [J]. 信息通信, 2013, 24 (9): 78-79.

[10] Dyabi M, Hajami A, Allali H. A new manets clustering algorithm based on nodes performances [C]// Proc of International Conference on Next Generation Networks & Services. New York: IEEE Press, 2014: 22-29.

[11] 唐李，刘改霞，钟汶娟，等. 移动 Ad hoc 网络多参数加权分簇算法 [J]. 重庆大学学报, 2014, 37 (2): 106-112.

[12] 马豫青，李晓宇. Ad hoc 网络自适应安全加权分簇算法 [J]. 计算机工程与设计, 2014, 9 (10): 3346-3350.

[13] Hu X, Wang J, Wang C. Stable clustering based on motion similarity evaluation in mobile Ad hoc networks [C]// Proc of International Symposium on Intelligent Information Technology Application. New York: IEEE Press, 2010: 234-237.

[14] Xu K, Hong X, Gerla M. An Ad hoc network with mobile backbones [C]// Proc of IEEE International Conference ON Communications. New York: IEEE Press, 2002: 3138-3143.

[15] 黄金科，樊晓光，万明，等. 基于稳定分簇的移动自组织网络路由协议 [J]. 北京航空航天大学学报, 2016, 42 (11): 2332-2339.

[16] Butler J, Olson D L. Comparison of centroid and simulation approaches for selection sensitivity analysis [J]. Journal of Multi-Criteria Decision Analysis, 2015, 8 (3): 146-161.

[17] Jia J, Fischer G W, Dyer J S. Attribute weighting method and decision quality in the presence of response error: a simulation study [J]. Journal of Behavioral Decision Making, 1998, 11 (2): 85-105.

[18] Stillwell W G, Seaver D A, Edwards W. A comparison of weight approximation techniques in multiattribute utility decision making [J]. Organizational Behavior & Human Performance, 1981, 28 (1): 62-77.